

GetSurDerivat.txt

功能：计算参数多项式曲面上参数为 (uparam, vparam) 的点处的关于u的urank阶关于v的vrank阶偏导矢，若 urank=vrank=0，则为曲面上点。

输入参数：uparam, vparam-曲面参数；urank, vrank-分别关于u、v的偏导矢阶数；mark-视图标识码：1, 2, 3-分别表示主、俯、左视图。SCx, SCy, SCz-曲面用幂基形式表示的系数矢量；m_uCount, m_vCount-分别为沿u、v数据点数；均是受保护成员。

输出参数：返回值p-偏导矢在mark表示的视图内的两分量。

调用函数：GetCuvDerivat-计算二维幂基多项式曲线导矢。

```
CPoint GetSurDerivat(double uparam, double vparam, int urank, int vrank, int mark)
{
    CArray<CArray<double, double>, CArray<double, double>&> temp;
    temp.SetSize(2);
    for(int i=0; i<2; i++) temp[i].SetSize(m_uCount);
    m_Cx.RemoveAll();
    m_Cy.RemoveAll();
    m_Cx.SetSize(m_vCount);
    m_Cy.SetSize(m_vCount);
    double temCx, temCy, px, py;
    CPoint point, p;
    for(int j=0; j<m_uCount; j++)
    {
        for(int i=0; i<m_vCount; i++)
        {
            switch(mark)
            {
                case 1:
                    temCx=SCx[i][j];
                    temCy=SCz[i][j];
                    break;
                case 2:
                    temCx=SCx[i][j];
                    temCy=SCy[i][j];
                    break;
                case 3:
                    temCx=SCy[i][j];
                    temCy=SCz[i][j];
                    break;
                default:;
            }
            m_Cx[i]=temCx;
            m_Cy[i]=temCy;
        }
        GetCuvDerivat(vrank, vparam, px, py);
        temp[0][j]=px;
        temp[1][j]=py;
    }
    m_Cx.RemoveAll();
    m_Cy.RemoveAll();
    m_Cx.SetSize(m_uCount);
    m_Cy.SetSize(m_uCount);
    for(j=0; j<m_uCount; j++)
    {
        m_Cx[j]=temp[0][j];
        m_Cy[j]=temp[1][j];
    }
    GetCuvDerivat(urank, uparam, px, py);
    p.x=int(px); p.y=int(py);
    return p;
}
```